

REMARKS/ARGUMENTS

Applicants thank the Examiner for his careful review of this application. Claims 1-26 have been rejected. Applicants respectfully request reconsideration of the application in view of the following remarks submitted in support thereof.

Missing Application Data in Specification

Applicants have amended the specification to include the missing application data. Accordingly, Applicants respectfully request the Examiner to withdraw the request for correction.

Obviousness Rejections of Claims 1-20 under 35 U.S.C. §103(a)

Claims 1-3, 5-6, and 8-20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,581,088 to Jacobs et al. in view of Kubiatowicz et al., *OceanStore: An Architecture for Global-Scale Persistent Storage* (2000). Claim 4 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Jacobs et al. in view of Kubiatowicz et al. and Deshpande et al., *Clustering: Transparent Replication, Load Balancing, and Failover* (2000). Claim 7 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Jacobs et al. in view of Kubiatowicz et al. and U.S. Patent No. 5,941,999 to Matena et al. As will be fully explained below, the combination of Jacobs et al. in view of Kubiatowicz et al. does not raise a *prima facie* case of obviousness against independent claims 1 and 11.

Independent claims 1 and 11 define a method and a system for migrating managed state for a Java based application. In this method and system, a first Java module is executed on a first server and the first Java module includes a first entity bean and a first state object in communication with the first entity bean. The first state object stores a state of the first entity

bean. Subsequently, the first state object is replicated to a state server and a second Java module is started on a second server, where the second Java module has a second state object.

In support of the obviousness rejection, the Examiner notes that the entity bean defined in independent claims 1 and 11 is disclosed as a smart stub in Jacobs et al. Applicants respectfully traverse the Examiner's characterization of Jacobs et al. relative to independent claims 1 and 11 because the portion of the reference relied upon by the Examiner (Abstract) does not teach or suggest the entity bean. In particular, Jacobs et al. explicitly disclose that "[s]tubs for stateless session beans are Smart stubs, while stubs for stateful session beans and *entity beans* are *not*" (col. 16, lines 63-65). Accordingly, Jacobs et al. unequivocally teach that an entity bean is simply not a smart stub.

Furthermore, the Examiner notes that Jacobs et al. teach or suggest executing a first Java module on a first server, and the first Java module includes a first entity bean and a first state object in communication with the first entity bean, where the first state object stores a state of the first entity bean. Applicants again respectfully traverse the Examiner's characterization of Jacobs et al. relative to independent claims 1 and 11 because the portion of the reference relied upon by the Examiner (column 4, lines 17-20) does not teach or suggest the first state object storing an entity bean. Specifically, assuming that the Examiner's assertion that a smart stub is an entity bean is true, then one would expect that Jacobs et al. at column 4, lines 17-20 would disclose something about a smart stub. However, column 4, lines 17-20 does not disclose anything about a smart stub, but instead discloses "a first software program emulating a processing device ("JVM1") including a first kernel software layer having a data structure ("RJVM1")" and a "second processing device includes a first software program emulating a processing device ("JVM2") including a first kernel software layer having a data structure ("RJVM2")." The JVM or Remote JVM "is a

program loaded onto a processing device which emulates a particular machine or processing device” (col. 2, lines 7-9). As a result, the JVM or RJVM is simply not an entity bean, and column 4, lines 17-20 does not disclose the first Java module including a first entity bean and a first state object in communication with the first entity bean, where the first state object stores a state of the first entity bean.

On the other hand, as discussed above, Jacobs et al. specifically disclose entity beans at column 16, lines 50-65. However, column 16, lines 50-65 does not anywhere teach or suggest a first Java module that includes a first entity bean and a first state object in communication with the first entity bean, where the first state object stores a state of the first entity bean, as defined in independent claims 1 and 11. Accordingly, Jacobs et al. cannot reasonably be considered to teach or suggest executing the first Java module on a first server, and the first Java module includes a first entity bean and a first state object in communication with the first entity bean, where the first state object stores a state of the first entity bean, as defined in independent claims 1 and 11.

To establish a *prima facie* case of obviousness, the prior art references must teach or suggest all the claim limitations (see M.P.E.P. §2143). Here, in view of the incorrect characterization of Jacobs et al., the references as combined do not teach all the features of the claimed invention.

Additionally, to establish a *prima facie* case of obviousness based on a combination of references, there must be some suggestion or motivation, either in the references or in the knowledge generally available to one having ordinary skill in the art, to combine the references in the manner proposed. The Examiner notes that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Jacobs et al. in view of Kubiatowicz et al. because “[o]ne would have been motivated to do so for durability and

availability of the state object ... Its deep archival storage mechanism permits information to survive in the face of global disaster” (see Office Action mailed July 22, 2004 at page 3).

The Applicants respectfully traverse the Examiner’s characterization of Jacobs et al. and Kubiatowicz et al. The teachings of Jacobs et al. focus on the “limited scalability or allowing for additional servers and/or clients” and “fault tolerance of redundant backend servers” (col. 3, lines 25-28). In contrast, the teachings of Kubiatowicz et al. focus on “replicat[ing] data for durability and availability” (page 192). Scalability of servers/clients and replicating data relate to entirely different applications. As the teachings of Jacobs et al. have nothing to do with the problem addressed by Kubiatowicz et al., Applicants submit that there would not have been any motivation for one having ordinary skill in the art to combine Jacobs et al. and Kubiatowicz et al. in the manner proposed by the Examiner.

Accordingly, for the above-stated reasons, Applicants submit that independent claims 1 and 11 are patentable under 35 U.S.C. §103(a) over Jacobs et al. in view of Kubiatowicz et al. Claims 2-10 and 12-20, each of which depends directly or indirectly from independent claims 1 and 11, are likewise patentable under 35 U.S.C §103(a) over Jacobs et al. in view of Kubiatowicz et al., Deshpande et al., and Matena et al. for at least the same reasons set forth for independent claims 1 and 11. As a result, Applicants respectfully request the Examiner to withdraw the 35 U.S.C. §103(a) rejection for claims 1-20.

Obviousness Rejections of Claims 21-26 under 35 U.S.C. §103(a)

Claims 21 and 23-26 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Silva et al., *Fault-Tolerant Execution of Mobile Agents* (2000) in view of Jacobs et al. Claim 22 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Silva et al. in view of Jacobs et al. and Kubiatowicz et al. As will be fully explained below, the

combination of Silva et al. in view of Jacobs et al. does not raise a *prima facie* case of obviousness against independent claim 21.

Independent claim 21 defines a method for initializing migrating managed state for a Java based application. Specifically, a request is sent to a second server to start a migrated module. The request is sent from a control module of a first module executing on a first server, where the control module passes a schema specification. Subsequently, the migrated module is created on the second server and a state of partitions for the migrated module is created based on the passed schema specification. A managed state is then recovered for the migrated module from a state server.

In support of the obviousness rejection, the Examiner notes that Jacobs et al. teach or suggest sending a request to a second server to start a migrated module, as defined in independent claim 21. Applicants respectfully traverse the Examiner's characterization of Jacobs et al. relative to independent claim 21 because the portion of the reference relied upon by the Examiner (column 1, lines 50-63) does not teach or suggest sending a request to a second server to start a migrated module. In particular, column 1, lines 56-61 of Jacobs et al. merely teach "a request from client 105" to "retrieve data from database 101a." In contrast, independent claim 21 defines a request to a second server to start a migrated module. A request to retrieve data is simply not the same as a request to start a migrated module. As Jacobs et al. disclose a different request, Jacobs et al. cannot reasonably be considered to teach or suggest sending a request to a second server to start a migrated module, as defined in independent claim 21.

Furthermore, the Examiner notes that Silva et al. teach or suggest creating a state partitions for the migrated module based on the passed schema specification. Applicants respectfully traverse the Examiner's characterization of Silva et al. relative to independent

claim 21 because the portion of the reference relied upon by the Examiner (Abstract) does not teach or suggest creating a state partitions for the migrated module based on the passed schema specification. In particular, Silva et al. disclose “a simple scheme to deal with network partitions” (Abstract). For example, “if the *execution* of the itinerary should be atomic the agent retries the migration to the nodes of the other partition.” Else, “[i]f the semantics is best-effort the agent reconfigures its itinerary and will try to *execute* in some other nodes of its partition” (Section 4.3.6). Accordingly, the scheme taught by Jacobs et al. to deal with network partitions involves the *execution* of mobile agents in the nodes of partitions. In contrast, independent claim 21 defines *creating* a state partitions for the migrated module. As the creation of state partitions is simply not the execution of mobile agents on nodes of partitions, Jacobs et al. cannot reasonably be considered to teach or suggest creating a state partitions for the migrated module based on the passed schema specification, as defined in independent claim 21.

To establish a *prima facie* case of obviousness, the prior art references must teach or suggest all the claim limitations (see M.P.E.P. §2143). Here, in view of the incorrect characterization of Jacobs et al. and Silva et al., the references as combined do not teach all the features of the claimed invention.

Additionally, to establish a *prima facie* case of obviousness based on a combination of references, there must be some suggestion or motivation, either in the references or in the knowledge generally available to one having ordinary skill in the art, to combine the references in the manner proposed. The Examiner notes that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Jacobs et al. in view of Silva et al. because “[o]ne would have been motivated to do so allow for enhanced scalability and fault tolerance” (see Office Action mailed July 22, 2004 at page 8). The Applicants

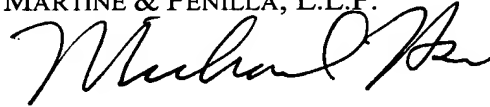
respectfully traverse the Examiner's characterization of Jacobs et al. and Silva et al. Again, the teachings of Jacobs et al. focus on the "limited scalability or allowing for additional servers and/or clients" and "fault tolerance of redundant backend servers" (col. 3, lines 25-28). In contrast, the teachings of Silva et al. focus on "fault-tolerance techniques that can increase the robustness of agent-based applications" (Abstract). Scalability of servers/clients and agent-based applications relate to entirely different applications. As the teachings of Jacobs et al. have nothing to do with the problem addressed by Silva et al., Applicants submit that there would not have been any motivation for one having ordinary skill in the art to combine Jacobs et al. and Silva et al. in the manner proposed by the Examiner.

Accordingly, for the above-stated reasons, Applicants submit that independent claim 21 is patentable under 35 U.S.C. §103(a) over Jacobs et al. in view of Silva et al. Claims 22-26, each of which depends directly or indirectly from independent claim 21, are likewise patentable under 35 U.S.C §103(a) over Silva et al. in view of Jacobs et al. and Kubiatowicz et al. for at least the same reasons set forth for independent claim 21. As a result, Applicant respectfully requests the Examiner to withdraw the 35 U.S.C. §103(a) rejection for claims 21-26.

Conclusion

In view of the foregoing, the Applicants respectfully submit that all the pending claims 1-26 are in condition for allowance. Accordingly, a Notice of Allowance is respectfully requested. If the Examiner has any questions concerning the present Amendment, the Examiner is requested to contact the undersigned at (408) 749-6900 ext. 6924. If any additional fees are due in connection with filing this Amendment, the Commissioner is also authorized to charge Deposit Account No. 50-0805 (Order No. SUNMP008). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
MARTINE & PENILLA, L.L.P.



Michael K. Hsu, Esq.
Reg. No. 46,782

Martine & Penilla, LLP
710 Lakeway Drive, Suite 170
Sunnyvale, California 94085
Telephone: (408) 749-6900
Customer Number 32,291